

## **LAPORAN MILESTONE 1 TUGAS BESAR IF 1210**



**Disusun Oleh : Kelompok L - K04**

Francis Galton	18224072
Rafael Sean Hadipranoto	18224100
Rashid May	18222014
Derren Christopher Boenardi	18224040
Sri Laksmi Purwoningtyas	18221009
Geodipa Afatha Ryu M.F.Z.	18224124

**MATA KULIAH ALGORITMA PEMROGRAMAN 1**  
**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA - KOMPUTASI**  
**INSTITUT TEKNOLOGI BANDUNG**  
**2025**

## HALAMAN PERNYATAAN

**Kami,**

Francis Galton	18224072
Rafael Sean Hadipranoto	18224100
Rashid May	18222014
Derren Christopher Boenardi	18224040
Sri Laksmi Purwoningtyas	18221009
Geodipa Afatha Ryu M.F.Z.	18224124

*"Kami menyatakan bahwa kami mengerjakan tugas besar ini dengan sejujur-jujurnya, tanpa menggunakan cara yang tidak dibenarkan. Apabila di kemudian hari diketahui kami mengerjakan tugas besar ini dengan cara yang tidak jujur, saya bersedia mendapatkan konsekuensinya, yaitu mendapatkan nilai E pada mata kuliah IF1210 Algoritma dan Pemrograman 1 Semester 2 2024/2025."*

:

## DAFTAR ISI

<b>HALAMAN PERNYATAAN.....</b>	<b>2</b>
<b>DAFTAR ISI.....</b>	<b>3</b>
<b>DAFTAR TABEL.....</b>	<b>5</b>
<b>DAFTAR GAMBAR.....</b>	<b>6</b>
<b>DESKRIPSI PERSOALAN.....</b>	<b>7</b>
<b>RENCANA IMPLEMENTASI.....</b>	<b>8</b>
<b>PEMBAGIAN KERJA.....</b>	<b>12</b>
<b>HASIL Pengerjaan.....</b>	<b>14</b>
<b>DESIGN COMMAND.....</b>	<b>15</b>
1. F01 - Login.....	15
2. F02 - Register.....	15
3. F03 - Logout.....	15
4. F04 - Lupa Password.....	15
5. F05 - Menu & Help.....	15
6. F06 - Denah Rumah Sakit.....	15
7. F10 - Tambah Dokter.....	16
8. F18 - Exit.....	16
<b>DESIGN KAMUS DATA.....</b>	<b>17</b>
1. F01 - Login.....	17
2. F02 - Register.....	17
3. F03 - Logout.....	17
4. F04 - Lupa Password.....	17
5. F05 - Menu & Help.....	17
6. F06 - Denah Rumah Sakit.....	17
7. F10 - Tambah Dokter.....	18
8. F18 - Exit.....	18
<b>DESIGN DEKOMPOSISI.....</b>	<b>19</b>
1. F01 - Login.....	19
2. F02 - Register.....	19
3. F03 - Logout.....	20
4. F04 - Lupa Password.....	20
5. F05 - Menu & Help.....	21
6. F06 - Denah Rumah Sakit.....	21
7. F10 - Tambah Dokter.....	22
8. F18 - Exit.....	22

<b>SPESIFIKASI MODUL.....</b>	<b>23</b>
1. Main Program.....	23
2. F01 - Login.....	25
3. F02 - Register.....	26
4. F03 - Logout.....	27
5. F04 - Lupa Password.....	27
6. F05 - Menu & Help.....	29
7. F06 - Denah Rumah Sakit.....	30
8. F10 - Tambah Dokter.....	31
9. F18 - Exit.....	35
<b>HASIL PENGUJIAN PROGRAM.....</b>	<b>36</b>
1. Inisialisasi Main Program.....	36
2. F01 - Login.....	36
3. F02 - Register.....	38
4. F03 - Logout.....	39
5. F04 - Lupa Password.....	39
6. F05 - Menu Help.....	40
7. F06 - Lihat Denah.....	42
8. F10 - Tambah Dokter.....	43
9. F18 - Exit.....	44
<b>LAMPIRAN.....</b>	<b>45</b>

## DAFTAR TABEL

Tabel 1.1 Rencana Implementasi.....	10
Tabel 2.1 Pembagian Kerja.....	14
Tabel 3.1 Hasil Pengerjaan.....	16

## DAFTAR GAMBAR

Gambar 6.1 Flowchart Login.....	19
Gambar 6.3 Flowchart Logout.....	20
Gambar 6.4 Flowchart Lupa Password.....	20
Gambar 6.5 Flowchart Menu & Help.....	21
Gambar 6.6 Flowchart Lihat Denah.....	21
Gambar 6.7 Flowchart Lihat Ruangan.....	22
Gambar 6.8 Flowchart Tambah Dokter.....	22
Gambar 6.9 Flowchart Exit.....	22
Gambar 7.1 Inisialisasi Main Program.....	36
Gambar 7.2 Login sebagai Manager.....	36
Gambar 7.2 Login sebagai Dokter.....	37
Gambar 7.3 Login sebagai Pasien.....	37
Gambar 7.4 Tidak Ada Username Terdaftar.....	37
Gambar 7.5 Kredensial Salah.....	38
Gambar 7.6 Register Berhasil.....	38
Gambar 7.7 Register Gagal.....	38
Gambar 7.8 Logout.....	39
Gambar 7.9 Password Berhasil Diubah.....	39
Gambar 7.10 User Tidak Terdaftar.....	40
Gambar 7.11 Kode Unik Salah.....	40
Gambar 7.12 Help Belum Login.....	40
Gambar 7.13 Help Manager.....	41
Gambar 7.14 Help Dokter.....	41
Gambar 7.15 Help Pasien.....	41
Gambar 7.16 Lihat Denah.....	42
Gambar 7.17 Lihat Ruangan.....	42
Gambar 7.18 Berhasil Tambah Dokter.....	43
Gambar 7.19 Tidak Berhasil Tambah Dokter.....	43
Gambar 7.18 Berhasil Assign Dokter ke Ruangan.....	43
Gambar 7.20 Dokter Sudah Diassign ke Ruangan Lain.....	44
Gambar 7.21 Ruangan Sudah Ditempati.....	44
Gambar 7.22 Ruangan Sudah Ditempati dan Dokter Sudah Diassign.....	44
Gambar 7.23 Exit.....	44

## **DESKRIPSI PERSOALAN**

Tugas besar ini meminta kami untuk membuat sebuah program sistem manajemen rumah sakit khusus yang dirancang untuk menangani pasien unik, yaitu para Nimon—makhluk kecil kuning yang sering mengalami kecelakaan karena tingkah mereka sendiri. Sistem ini bertujuan untuk membantu Dr. Neroifa dalam mengelola rumah sakit baru yang ia bangun bersama Gro dan Luiy, setelah menyadari bahwa para Nimon sering terluka dan belum ada sistem medis yang tertata dengan baik.

Dalam program ini, akan terdapat berbagai fungsionalitas seperti proses login dan registrasi pasien, melihat denah rumah sakit, menambah dokter, mendiagnosis penyakit, memberikan obat, serta mengelola antrian pemeriksaan pasien. Pasien dapat mendaftar check-up, mengonsumsi obat, meminum penawar jika salah obat, dan berkonsultasi untuk pulang. Di sisi lain, dokter dapat menangani diagnosis dan pengobatan, sementara manajer dapat mengelola data pengguna dan melihat semua antrian. Kami diminta menggunakan berbagai struktur data seperti Set, Map, Stack, Queue, dan List untuk menyimpan dan memproses data secara efisien.

Dengan menyelesaikan program ini, kami diharapkan mampu mengorganisasi kekacauan yang ada menjadi sistem rumah sakit digital yang rapi, fungsional, dan dapat membantu semua pengguna—baik pasien, dokter, maupun manajer—dalam aktivitas medis di dunia para Nimon.

## RENCANA IMPLEMENTASI

Tabel 1.1 Rencana Implementasi

Implementasi ADT	FITUR	Deskripsi Implementasi	Alasan Implementasi
ADT Set, ADT List Statis	F01 - Login	<p>UserSet menyimpan seluruh data user dalam bentuk array, serta memastikan tidak ada duplikasi username.</p> <p>UserSet menggunakan array statis ‘User users[MAX_USER S]’ sebagai penyimpanan data user, sehingga pengaksesan cepat dan terstruktur.</p>	<p>Penggunaan ADT Set dan List Statis memudahkan validasi data login dengan cepat karena pencarian dilakukan dalam memori. Selain itu, struktur ini sesuai untuk sistem dengan jumlah user terbatas dan memperjelas batasan sistem (maksimal MAX_USERS). Hal ini juga mendukung modularitas dan pengujian program tanpa tergantung file eksternal.</p>
ADT Set, ADT List Statis	F02 - Register	<p>UserSet menyimpan data user baru yang berisikan username password dan role ke dalam array statis ‘User users[MAX_USER S]’</p> <p>Saat melakukan registrasi, sistem</p>	<p>Penggunaan ADT Set (untuk memastikan keunikan username) dan List Statis (untuk penyimpanan data user dalam array) memudahkan manajemen user tanpa perlu alokasi dinamis atau file</p>



		<p>akan memeriksa apakah username sudah ada atau belum menggunakan linear search dan menambahkan user baru ke array jika tidak ada. Fitur ini memastikan tidak ada duplikasi username dan jumlah user tidak melebihi batas maksimum</p>	<p>eksternal. Hal ini sesuai untuk sistem dengan jumlah user terbatas dan membuat pengujian lebih mudah dan cepat.</p>
ADT User Sederhana	F03 - Logout	<p>mengecek 'current_user' apakah memiliki isi, dan kemudian mengkosongkannya lagi menjadi NULL.</p>	<p>Logout cukup dengan mengatur ulang pointer current_user tanpa perlu menghapus atau mengubah data user, sehingga implementasi tetap efisien dan sederhana. ADT User menyimpan semua informasi login, sehingga pointer ke objek ini cukup untuk menjaga konteks pengguna selama sesi berlangsung.</p>
ADT List Statik	F04 - Lupa Password	<p>Digunakan untuk mencari user di dalam 'UserSet' menggunakan fungsi 'findUser'.</p> <p>Digunakan untuk membandingkan kode unik hasil inputan user dengan rleCode hasil</p>	<p>Karena jumlah maksimum untuk user (MAX_USER) dan panjang passwordnya (MAX_LEN) sudah di-define, sehingga tidak memerlukan array dinamis. Pencarian user dengan data set</p>

		'generateRLE'	yang relatif kecil ini lebih cepat dilakukan dengan pencarian linear melalui indeks. Penggunaan ADT List Statik ini juga sudah memenuhi kriteria dan lebih aman untuk dipakai karena tidak ada alokasi memori.
ADT User Sederhana	F05 - Menu & Help	Mengecek apakah <code>current_user</code> menunjuk pada user yang login. Jika tidak (NULL), maka tampilkan menu bantuan dasar. Jika ya, tampilkan menu sesuai <code>current_user-&gt;role</code> .	ADT User menyimpan informasi peran (role) sehingga cukup dengan memeriksa <code>current_user</code> untuk menentukan hak akses dan jenis bantuan yang relevan.
ADT List Statis, ADT Map, dan ADT Set	F06 - Denah Rumah Sakit	Digunakan untuk mengakses jumlahRuangan dari fitur LIHAT_DENAH  Digunakan ADT List, Map, dan Set dalam fitur LIHAT_RUANGA N	Menggunakan ADT yang dapat merepresentasikan kumpulan ruangan dalam denah  Menggunakan ADT list untuk validasi inputan id Ruangan dan mengambil kapasitas dan informasi pasien dari elemen ruangan. Selain itu, digunakan ADT Map dalam pembuatan logika <i>pairing</i> antara idDokter dan

			idRuangan. Mengambil detail username dokter dan pasien menggunakan ADT Set.
ADT Set, ADT Map, ADT List Statis	F10 - Tambah Dokter	<p>Menerima input untuk dokter baru dan menggunakan operasi ADT Set dalam register pada command TAMBAH_DOKTER</p> <p>Digunakan ADT Set, List, dan MAP pada command ASSIGN DOKTER</p>	<p>Karena cocok untuk mengelola <i>unique pairing</i> dari informasi pengguna.</p> <p>Tugas utama dari fitur ini yaitu membuat pemetaan untuk assign dokter ke ruangan dan didukung ADT Set dan List sebagai validator.</p>

## PEMBAGIAN KERJA

Tabel 2.1 Pembagian Kerja

Fitur	Implementasi *)	NIM Desainer **)	NIM Coder **)	NIM Tester **)
F01 - Login	procedure loginUser(), ADT List Statik, ADT set	18224100 18224124	18224100 18224124	18222009 18224040 18224100 18224124
F02 - Register	procedure registerUser(), ADT List Statik, ADT set	18224124	18224124	18222009 18224040 18224100 18224124
F03 - Logout	procedure logoutUser(), ADT List	18224100	18224100	18222009 18224040 18224100 18224124
F04 - Lupa Password	procedure generateRLE(), function findUser(), procedure registerUser(), procedure lupaPassword()	18224040	18224040	18222009 18224040 18224100 18224124
F05 - Menu & Help	procedure help(), ADT user sederhana	18222014	18222014	18222014 18221009
F06 - Denah Rumah Sakit	ADT List Statik, procedure initDenah(), function getPanjangDena h(), function getKapasitasDen ah(), procedure lihatDenah(), procedure lihatRuangan()	18221009	18221009	18221009
F10 - Tambah Dokter	ADT List Statik, ADT Map, procedure initMapDokterRu	18221009	18221009	18221009

	angan(), function bool assignDokterToR uangan(), function bool unassignDokter() , function getRuanganByDo kter(), function getDokterByRuan gan(), function isDokterAssigne dtoRuangan(), function isRuanganAssign edToDokter(), procedure managerTambah Dokter(), procedure assignDokter()			
F18 - Exit	-	18224072	18224072	18221009 18224072

## HASIL Pengerjaan

Tabel 3.1 Hasil Pengerjaan

Fitur	Desain	Implementasi	Testing
F01 - Login	V	V	V
F02 - Register	V	V	V
F03 - Logout	V	V	V
F04 - Lupa Password	V	V	V
F05 - Menu & Help	V	V	V
F06 - Denah Rumah Sakit	V	V	V
F10 - Tambah Dokter	V	V	V
F18 - Exit	V	V	V

**Keterangan:** V: sudah selesai dikerjakan, X: dikerjakan, tapi belum selesai, -: tidak dikerjakan sama sekali.

## DESIGN COMMAND

### 1. F01 - Login

Command : Login

Masukan(I.S.): Data UserSet (user yang terdaftar) sudah ada, input username dan password.

Keluaran(F.S.): Apabila sesuai dengan yang terdaftar “Login Berhasil”, apabila tidak sesuai “Login Gagal”.

### 2. F02 - Register

Command : Register

Masukan : Mengisi Data UserSet (Username, Password, Role)

Keluaran : Apabila sudah ada yang terdaftar, “Akun sudah teregistrasi”. apabila tidak, “Akun berhasil di registrasi”

### 3. F03 - Logout

Command : Logout

Masukan(I.S.) : Sudah login sebelumnya. Meminta untuk logout.

Keluaran(F.S.) : Ter-Logout dari user tersebut.

### 4. F04 - Lupa Password

Command : Lupa Password

Masukan(I.S.) : Data UserSet berisi user yang terdaftar, proses pembaharuan password belum dijalankan.

Keluaran(F.S.) : Jika user ditemukan dan rleCode sesuai, password user diperbaharui ; Jika tidak sesuai, tidak ada perubahan dalam UserSet

### 5. F05 - Menu & Help

Command : Menu & Help

Masukan : Pengguna berada dalam kondisi login atau belum login.

Keluaran : Sistem menampilkan daftar perintah yang dapat dijalankan sesuai status login dan peran pengguna.

### 6. F06 - Denah Rumah Sakit

Command : Denah Rumah Sakit

Masukan : Denah yang di-generate ketika menjalankan main program, idRuang yang ingin dilihat, set User yang di-generate ketika menjalankan main program, dan map dokter.

Keluaran : Sistem menampilkan denah yaitu banyak ruangan yang diinisialisasi sebelumnya serta detail tiap ruangan (kapasitas ruangan, dokter, dan pasien dalam ruangan)

**7. F10 - Tambah Dokter**

Command : Tambah Dokter

Masukan : idDokter dan idRuangan

Keluaran : Sistem melakukan validasi register untuk menambah dokter yang bertugas ke dalam program oleh manager. Lalu, sistem melakukan assign dokter ke ruangan.

**8. F18 - Exit**

Command : Exit

Masukan : -

Keluaran : Pesan berhasil keluar atau Kembali ke menu awal



# DESIGN KAMUS DATA

## 1. F01 - Login

```
type User : <id : int
            username : string,
            password : string,
            role : enum {MANAGER, DOKTER, PASIEN}>

type UserSet : < users : array [0..MAX_USERS] of User,
                jumlahUser : integer >
current_user : pointer to User
```

## 2. F02 - Register

```
username, password, role, user_baru, access : string
data_user : array of strings
index, jumlahUser : integer
username_registered : boolean
```

## 3. F03 - Logout

```
current_user : pointer to User
```

## 4. F04 - Lupa Password

```
username, rleCode, newPassword, correctRle : string
user : pointer to User
```

## 5. F05 - Menu & Help

```
current_user : pointer to User
username : string
role : enum {MANAGER, DOKTER, PASIEN}
```

## 6. F06 - Denah Rumah Sakit

```
type Ruang : < idDokter : integer,
                idPasien array of [0..MAX_PASIEN] : integer,
                jumlahPasien : integer,
                idRuang : integer,
                antrianPasien : Queue >
```

```
type Denah : < ruangan array of [0..MAX_RUANGAN] : Ruangan,  
                jumlahRuangan : integer,  
                kapasitas : integer >
```

## 7. F10 - Tambah Dokter

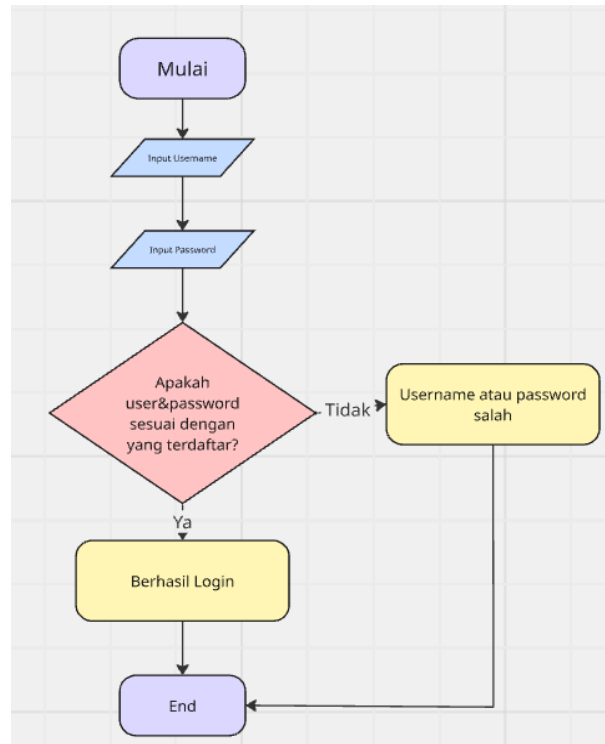
```
type DokterRuangan : < idDokter : integer,  
                        idRuangan : integer >  
  
type MapDokterRuangan : < map array of DokterRuangan  
[0..MAX_DOKTER],  
                        jumlahDokter : integer >
```

## 8. F18 - Exit

```
input : char
```

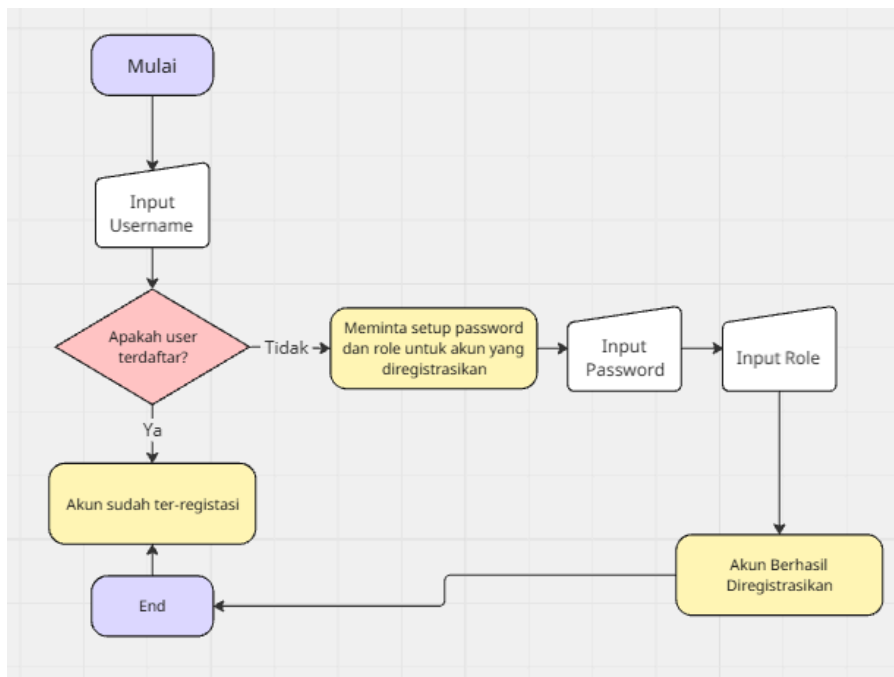
## DESIGN DEKOMPOSISI

### 1. F01 - Login



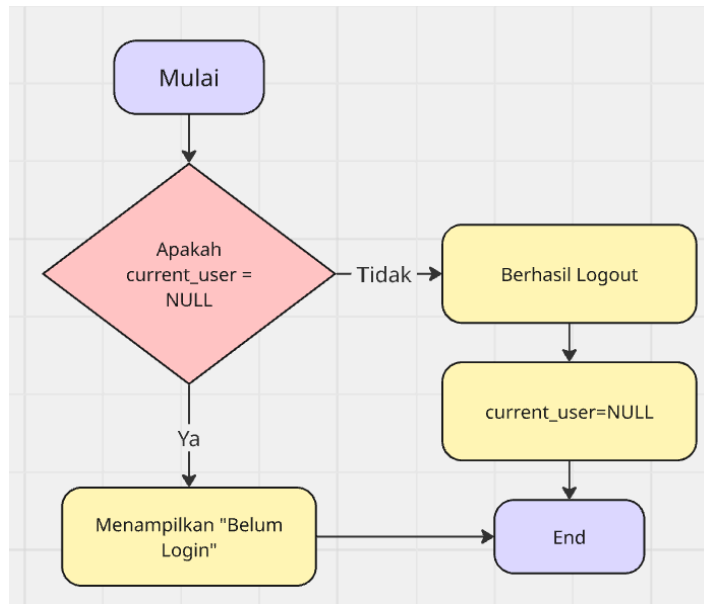
Gambar 6.1 Flowchart Login

### 2. F02 - Register



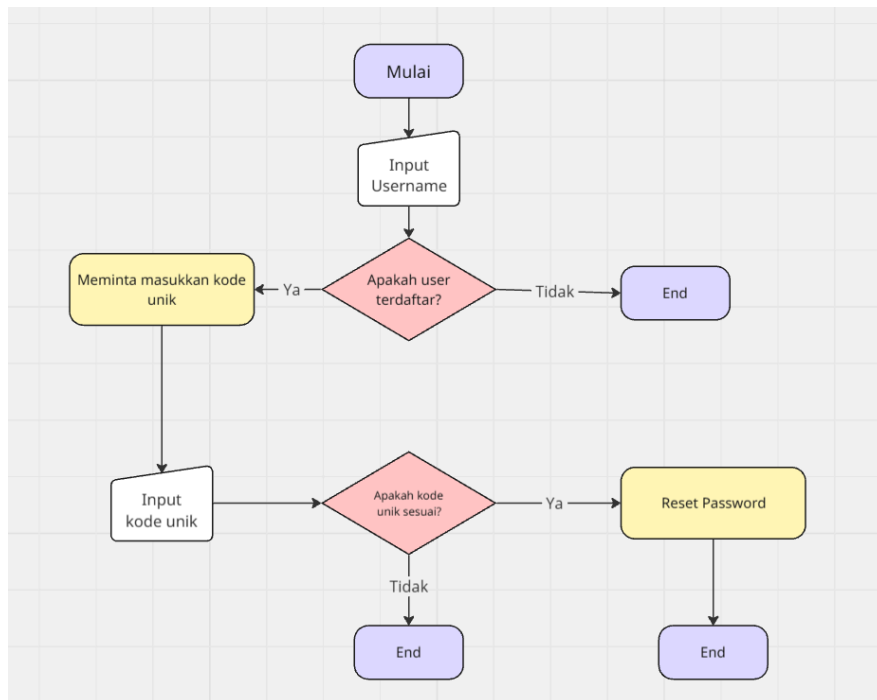
Gambar 6.2 Flowchart Register

### 3. F03 - Logout



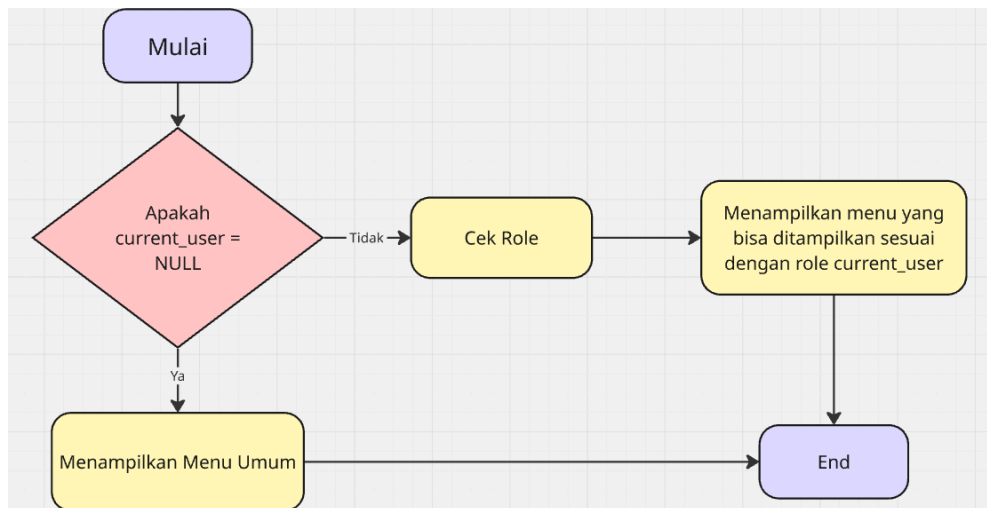
Gambar 6.3 Flowchart Logout

### 4. F04 - Lupa Password



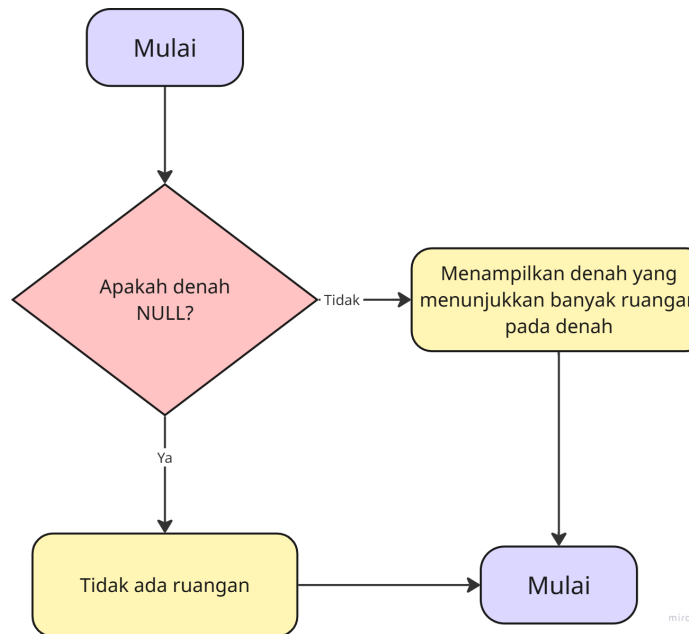
Gambar 6.4 Flowchart Lupa Password

## 5. F05 - Menu & Help

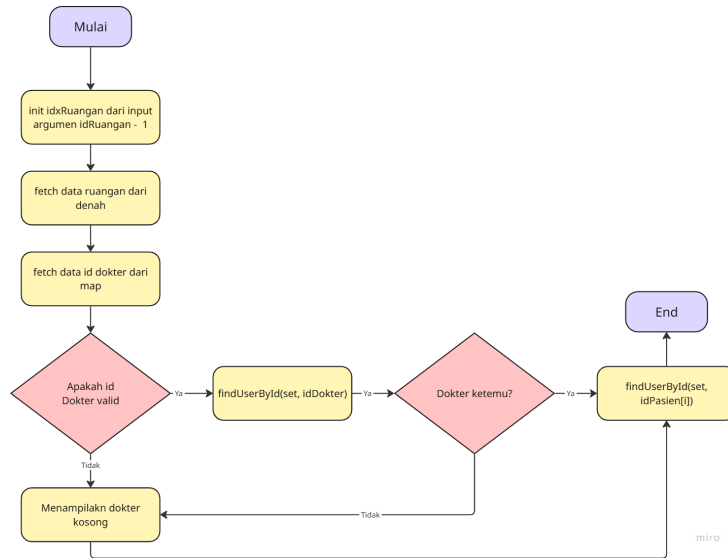


Gambar 6.5 Flowchart Menu & Help

## 6. F06 - Denah Rumah Sakit

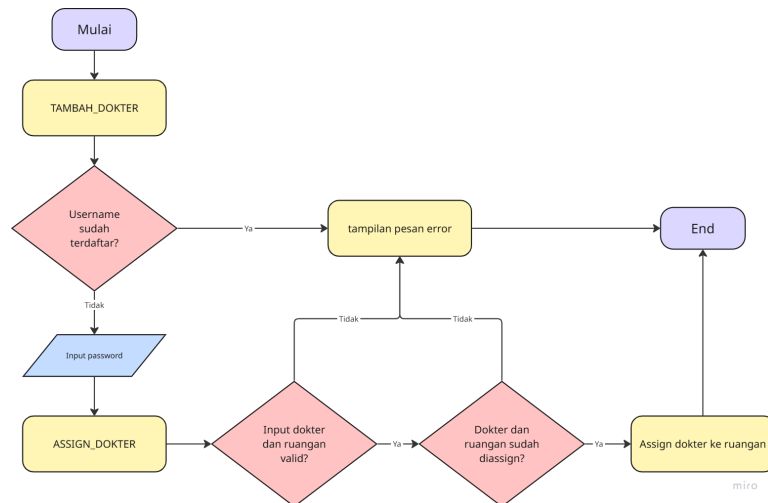


Gambar 6.6 Flowchart Lihat Denah



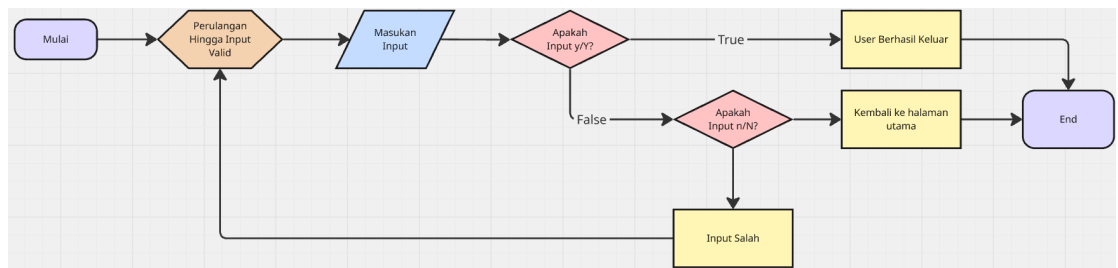
Gambar 6.7 Flowchart Lihat Ruangan

## 7. F10 - Tambah Dokter



Gambar 6.8 Flowchart Tambah Dokter

## 8. F18 - Exit



Gambar 6.9 Flowchart Exit

# SPESIFIKASI MODUL

## 1. Main Program

```
procedure tampilanBanner()
    output ""
    output "+-----+"
    output "|                SELAMAT DATANG                |"
    output "|                DI RUMAH SAKIT NIMONS            |"
    output "+-----+"
    output "                powered by Kelompok L          "
    output "                K04 - STI                      "
    output ""

procedure tampilkanMenuUtama()
    if current_user = null then
        output ""
        output "Selamat datang! Silakan pilih opsi:"
        output "1. LOGIN"
        output "2. REGISTER PASIEN"
        output "4. LUPA PASSWORD"
        output "5. HELP"
        output "0. KELUAR"
    else
        output ""
        output "Halo, " + roleToStr(current_user.role) + " " +
current_user.username + "!"
        output "-----"
        output "Menu yang Tersedia:"
        output "1. LIHAT DENAH"
        output "2. LIHAT RUANGAN"
        output "5. HELP "

        if current_user.role = MANAGER then
            output "10. TAMBAH DOKTER"
            output "11. ASSIGN DOKTER"

            output "9. LOGOUT "
            output "0. EXIT"
        output "-----"
        output "Pilihan Anda: "

program utama
    tampilanBanner()

    initUserSet(generateDataPengguna)
    initMapDokterRuangan(generateDataMap)

    define jumlahRuangan, kapasitas as integer
    output "Masukkan jumlah ruangan: "
    input jumlahRuangan
    if input_gagal_atau_bukan_integer then
        output "Input tidak valid. Masukkan jumlah ruangan."
        → 1
```

```

clearInputBuffer()

output "Masukkan kapasitas ruangan: "
input kapasitas
if input_gagal_atau_bukan_integer then
    output "Input tidak valid. Masukkan kapasitas ruangan."
    → 1
clearInputBuffer()

initDenah(generateDataDenah, jumlahRuangan, kapasitas)

define pilihan as integer
define usernameInput as string[MAX_LEN]
define passwordInput as string[MAX_LEN]
define idRuanganInput as integer

repeat
    tampilkanMenuUtama()
    input pilihan
    if input_gagal_atau_bukan_integer then
        output "Input tidak valid. Masukkan nomor pilihan."
        clearInputBuffer()
        pilihan ← -1
        continue 1
    clearInputBuffer()

    if current_user = null then -- Belum Login
        switch pilihan
            case 1:
                output ">>> LOGIN"
                output "Username : "
                input usernameInput
                output "Password : "
                input passwordInput
                loginUser(generateDataPengguna, usernameInput,
passwordInput)
            case 2:
                output ">>> REGISTER"
                output "Username : "
                input usernameInput
                output "Password : "
                input passwordInput
                registerUser(generateDataPengguna, usernameInput,
passwordInput, PASIEN)
            case 4:
                lupaPassword(generateDataPengguna)
            case 5:
                output ">>> HELP"
                help()
            case 0:
                keluar()
            default:
                output "Pilihan tidak valid."

        else -- Sudah Login
            switch pilihan

```



```

        case 1:
            lihatDenah(generateDataDenah)
        case 2:
            output "Masukkan ID Ruangan yang ingin dilihat : "
            input idRuanganInput
            if input_berhasil_dan_integer then
                clearInputBuffer()
                lihatRuangan(generateDataDenah, idRuanganInput,
generateDataPengguna, generateDataMap)
            else
                output "Input ID Ruangan tidak valid."
                clearInputBuffer()
            endif
        case 5:
            output ">>> HELP"
            help()
        case 10:
            if current_user.role = MANAGER then
                managerTambahDokter(generateDataPengguna)
            else
                output "Anda tidak memiliki akses untuk fitur ini."
            endif
        case 11:
            if current_user.role = MANAGER then
                assignDokter(generateDataMap, generateDataDenah,
generateDataPengguna)
            else
                output "Anda tidak memiliki akses untuk fitur ini."
            endif
        case 9:
            logoutUser()
        case 0:
            keluar()
        default:
            output "Pilihan tidak valid."
    until pilihan = 0

    → 0

```

## 2. F01 - Login

```

function loginUser(set : UserSet, username : string, password :
string) → User
DEKLARASI
    user_found : User
    i : integer

DESKRIPSI
    input (username)
    input (password)
    user_found ← findUserByUsername(set, username)

    if user_found = NULL then
        output("Tidak ada Manager, Dokter, atau pun Pasien yang
bernama ", username)
        return NULL
    endif

```

```

endif

for i ← 0 HINGGA set.jumlahUser - 1 fo
    if set.users[i].username = username && set.users[i].password
= password then
        current_user ← set.users[i]
        output("Login berhasil. Halo, ", username)

        if current_user.role = MANAGER then
            output("Selamat pagi Manager ", current_user.username)
        else if current_user.role = DOKTER then
            output("Selamat pagi Dokter ", current_user.username)
        else if current_user.role = PASIEN then
            output("Selamat pagi Pasien ", current_user.username, "!
Ada keluhan apa?")
        endif

        return current_user
    endif
endfor

output("Login gagal. Username atau password salah.")
return NULL

```

### 3. F02 - Register

```

procedure registerUser (input username : string, input password :
string, input role : enum {MANAGER,DOKTER,PASIEN}, in/out set :
UserSet)

Deklarasi:
i : integer
usernameRegistered : boolean
newUser : User

For i ← 0 set.jumlahUser - 1 do
    if set.users[i].username = username then
        usernameRegistered ← true
        print "Username sudah terdaftar"
        return
    endif
endFor

if set.jumlahUser = MAX_USERS then
    print "User limit reached"
    return
endif

newUser.id ← set.jumlahUser + 1
newUser.username ← username
newUser.password ← password
newUser.role ← role

```

```
set.users[set.jumlahUser] ← newUser
set.jumlahUser ← set.jumlahUser + 1

Print "Registrasi berhasil untuk ", username
```

#### 4. F03 - Logout

```
function logoutUser() → boolean

    output(">>> LOGOUT")

    if current_user != NULL then
        output("Logout berhasil. Sampai jumpa, ",
current_user.username)
        current_user ← NULL
        return true
    endif

    output("Belum login.")
    return false
```

#### 5. F04 - Lupa Password

```
DEKLARASI
    username : string
    password : string
    role : string
    userCount : integer
    MAX_USERS : integer
    users : array of User

ALGORITMA
procedure registerUser(username : string, password : string, role
: string)
    if (userCount < MAX_USERS) then
        users[userCount].username ← username
        users[userCount].password ← password
        users[userCount].role ← role
        userCount ← userCount + 1
```

```
DEKLARASI
    username : string
    rleCode : string
    correctRle : string
    newPassword : string
    user : pointer to User

ALGORITMA
procedure lupaPassword()
    output(">>> LUPA_PASSWORD")
```

```

output("Username: ")
input(username)

user ← findUser(username)
if (user = NULL) then
    output("Username tidak terdaftar!")
    return

output("Kode Unik: ")
input(rleCode)

generateRLE(user.username, correctRle)

if (rleCode ≠ correctRle) then
    output("Kode unik salah!")
    return

output("Halo", user.role, user.username, ", silakan daftarkan
ulang password anda!")
output("Password Baru: ")
input(newPassword)

user.password ← newPassword
output("Password berhasil diubah!")

```

#### DEKLARASI

```

username : string
user : pointer to User
i : integer

```

#### ALGORITMA

```

procedure findUser(username : string) → pointer to User
    user ← NULL
    i ← 0

    while (i < userCount) do
        if (users[i].username = username) then
            user ← address of users[i]
            return user

        i ← i + 1

    return NULL

```

#### DEKLARASI

```

username : string
rleCode : string
len : integer
count : integer
i : integer
rleIndex : integer

```

#### ALGORITMA

```

procedure generateRLE(username : string, output rleCode : string)
    len ← panjang(username)

```

```

count ← 1
rleIndex ← 0

for i ← 1 to len do
    if (username[i] = username[i - 1]) and (i < len) then
        count ← count + 1
    else
        if count > 1 then
            append(count) to rleCode
        append(username[i - 1]) to rleCode
        count ← 1

append('\0') to rleCode

```

## 6. F05 - Menu & Help

```

procedure help

DEKLARASI
    current_user : Pointer to User
    username : string
    role : enum {MANAGER, DOKTER, PASIEN}

    if (current_user = NULL) then
        output("Kamu belum login sebagai role apapun. Silahkan
login terlebih dahulu.")
        output("")
        output("1. LOGIN: Masuk ke dalam akun yang sudah
terdaftar")
        output("2. REGISTER: Membuat akun baru")
    else
        case (current_user→role) of
            MANAGER:
                output("Halo Manager ", current_user→username, ".
Kenapa kamu memanggil command HELP? Kan kamu manager, tapi
yasudahlah kamu pasti sedang kebingungan. Berikut adalah hal-hal
yang dapat kamu lakukan:")
                output("1. LOGOUT: Keluar dari akun")
                output("2. TAMBAH_DOKTER: Menambahkan dokter
baru")
                output("3. LIHAT_USER: Melihat seluruh user")
                output("4. EXIT: Keluar dari sistem")

            DOKTER:
                output("Halo Dokter ", current_user→username, ".
Kamu memanggil command HELP. Kamu pasti sedang kebingungan.
Berikut adalah hal-hal yang dapat kamu lakukan:")
                output("1. LOGOUT: Keluar dari akun")
                output("2. DIAGNOSIS: Melakukan diagnosis
pasien")
                output("3. NGOBATIN: Memberikan obat")
                output("4. EXIT: Keluar dari sistem")

```

```

PASIEN:
    output("Selamat datang, ", current_user->username,
". Kamu memanggil command HELP. Kamu pasti sedang kebingungan.
Berikut adalah hal-hal yang dapat kamu lakukan:")
    output("1. LOGOUT: Keluar dari akun")
    output("2. DAFTAR_CHECKUP: Daftar untuk
pemeriksaan")
    output("3. MINUM_OBAT: Minum obat")
    output("4. PULANGDOK: Meminta izin pulang")
    output("5. EXIT: Keluar dari sistem")

    output("")
    output("Footnote:")
    output("1. Untuk menggunakan aplikasi, silahkan masukkan nama
fungsi yang terdaftar")
    output("2. Jangan lupa untuk memasukkan input yang valid")

```

## 7. F06 - Denah Rumah Sakit

```

procedure initDenah (input/output denah: Denah*, input
jumlahRuangan : integer, input kapasitas : integer)

if denah = NULL then
->

if jumlahRuangan <0 OR kapasitas <0 then
    denah.jumlahRuangan <- 0
->

denah.jumlahRuangan <- jumlahRuangan
denah.kapasitas <- kapasiras

i traversal [0..jumlahRuangan - 1]
    denah.ruangan[i].idDokter <- ID_INVALID
    denah.ruangan[i].idRuangan <- ID_INVALID
    denah.ruangan[i].jumlahPasien <- 0
    j traversal {0..MAX_PASIEN-1} do
        denah.ruangan[i].idPasien[j] <- ID_INVALID
    initQueue(denah.ruangan[i].antrianPseien)

function getPanjangDenah(denah : Denah*) -> integer
if denah = NULL then
-> 0
-> denah.jumlahRuangan

function getKapasitasDenah (denah: Denah*) -> integer
if denah = NULL then
-> 0
-> denah.kapasitas

procedure lihatDenah(input/output denah : Denah*)
if denah = NULL or denah.jumlahRuangan = 0 then
    outout "tidak ada ruangan"

```

```

->

i traversal [0..denah.jumlahRuangan - 1]
  outout "+-----"

i traversal [0..denah.jumalhRuangan-1]
  output"| " + (i+1) " |"

i traversal [0..denah.jumlahRuangan - 1]
  outout "+-----"

procedure lihatRuangan(input/output denah: Denah*, input
idRuangan : integer, set : UserSet, map : mapDokterRuangan*)

idxRuangan ← idRuangan - 1
ruang ← denah.ruangan[idxRuangan]
output ">>> LIHAT_RUANGAN " + idRuangan
output ""
output "--- Detail Ruangan " + idRuangan + " ---"
output "Kapasitas : " + denah.kapasitas

idDokterDiRuangan ← getDokterByRuangan(map, idRuangan)
output "Dokter      : "
if idDokterDiRuangan ≠ ID_INVALID then
  dokter ← findUserById(set, idDokterDiRuangan)

  if dokter ≠ NULL then
    output "Dokter " + dokter.username
  else
    output "-"
ruangPasien ← denah.ruangan[idxRuangan]
outoput "Pasien di dalam ruangan : "
if ruangPasien.jumlahPasien > 0 then
  i traversal [0..ruangPasien.jumlahPasien - 1]
    pasien ← findUserById(set, ruangPasien.idPasien[i])
    if pasien ≠ NULL then
      outout " " + (i + 1) + ". " + pasien.username
else
  outout " Tidak ada pasien di dalam ruangan saat ini."
output "-----"

```

## 8. F10 - Tambah Dokter

```

procedure initMapDokterRuangan(input/output map:
MapDokterRuangan*)

  if map = null then
    return

  map.jumlahDokter ← 0
  i traversal [0..MAX_DOKTER - 1]
    map.map[i].idDokter ← ID_INVALID
    map.map[i].idRuangan ← ID_INVALID

```

```

function assigDokterToRuangan(map: MapDokterRuangan*, idDokter:
integer, idRuangan: integer) -> boolean
  if map = null or idDokter = ID_INVALID or idRuangan =
ID_INVALID then
    -> false

  if map.jumlahDokter ≥ MAX_DOKTER then
    -> false

  if isDokterAssignedToRuangan(map, idDokter) then
    -> false

  if isRuanganAssignedTDokter(map, idRuangan)
    -> false

  map.map[map.jumlahDokter].idDokter ← idDokter
  map.map[map.jumlahDokter].idRuangan ← idRuangan
  map.jumlahDokter ← map.jumlahDokter + 1

  -> true

```

```

function unassignDokterFromRuangan( map: MapDokterRuanganPointer,
idDokter: integer) -> boolean
  if map = null or idDokter = ID_INVALID then
    -> false
  idxFound ← -1
  i travesak [0..map.jumlahDokter - 1]
    if map.map[i].idDokter = idDokter then
      idxFound ← i
      break

  if idxFound ≠ -1 then
    if idxFound < map.jumlahDokter - 1 then
      map.map[idxFound] ← map.map[map.jumlahDokter - 1]
      map.map[map.jumlahDokter - 1].idDokter ← ID_INVALID
      map.map[map.jumlahDokter - 1].idRuangan ← ID_INVALID
      map.jumlahDokter ← map.jumlahDokter - 1
    -> true
  else
    -> false

```

```

function getRuanganByDokter( map: MapDokterRuanganPointer,
idDokter: integer) -> integer
  if map = null or idDokter = ID_INVALID then
    -> ID_INVALID

  i traversal [0..map.jumlahDokter - 1]
    if map.map[i].idDokter = idDokter then
      -> map.map[i].idRuangan

  -> ID_INVALID

```

```

function getDokterByRuangan( map: MapDokterRuanganPointer,
idRuangan: integer) -> integer
  if map = null or idRuangan = ID_INVALID then
    -> ID_INVALID

```



```
i traversal [0..map.jumlahDokter - 1 ]
  if map.map[i].idRuangan = idRuangan then
    -> map.map[i].idDokter

-> ID_INVALID
```

```
function isDokterAssignedToRuangan( map: MapDokterRuanganPointer,
idDokter: integer) -> boolean
  -> getRuanganByDokter(map, idDokter) ≠ ID_INVALID
```

```
function isRuanganAssignedToDokter( map: MapDokterRuanganPointer,
idRuangan: integer) -> boolean
  -> getDokterByRuangan(map, idRuangan) ≠ ID_INVALID
```

```
procedure managerTambahDokter(input/output set: UserSetPointer)
  if set = null then
    →

  define username as string[MAX_LEN]
  define password as string[MAX_LEN]

  output ">>> TAMBAH_DOKTER"

  output "Username: "
  input username
  if input_gagal then
    output "Gagal membaca input."
    →

  existingUser ← findUserByUsername(set, username)
  if existingUser ≠ null then
    if existingUser.role = DOKTER then
      output "Username " + username + " sudah terdaftar sebagai
dokter."
    else
      output "Username " + username + " sudah terdaftar."
    →

  output "Password: "
  input password
  if input_gagal then
    output "Gagal membaca input."
    →

  dokterBaru ← registerUser(set, username, password, DOKTER)
```

```
procedure assignDokter(input/output map: MapDokterRuanganPointer,
input denah: DenahPointer, input set: UserSetPointer)

  define username as string[MAX_LEN]
  define idRuangan as integer

  output ">>> ASSIGN_DOKTER"
  output "Username: "
  input username
```

```

if input_gagal then
    output "Gagal membaca input."
    →

output "ID Ruangan : "
input idRuangan

dokter ← findUserByUsername(set, username)
if dokter = null or dokter.role ≠ DOKTER then
    output "Dokter dengan username " + username + " tidak
ditemukan."
    →

if idRuangan < 1 or idRuangan > denah.jumlahRuangan then
    output "Ruangan dengan ID " + idRuangan + " tidak ditemukan."
    →

ruanganSekarang ← getRuanganByDokter(map, dokter.id)
dokterSekarangId ← getDokterByRuangan(map, idRuangan)
dokterDiRuanganItu ← findUserById(set, dokterSekarangId)

define namaDokterDiRuanganItu as string
if dokterDiRuanganItu ≠ null then
    namaDokterDiRuanganItu ← dokterDiRuanganItu.username
else
    namaDokterDiRuanganItu ← ""

if ruanganSekarang ≠ ID_INVALID and dokterSekarangId ≠
ID_INVALID then
    output ""
    output "Dokter " + dokter.username + " sudah ada di ruangan "
+ ruanganSekarang + "!"
    output "Ruangan " + idRuangan + " juga sudah ditempati dokter
" + namaDokterDiRuanganItu + "!"
    →

if ruanganSekarang ≠ ID_INVALID then
    output ""
    output "Dokter " + dokter.username + " sudah diassign ke
ruangan " + ruanganSekarang + "!"
    →

if dokterSekarangId ≠ ID_INVALID then
    output ""
    output "Dokter " + namaDokterDiRuanganItu + " sudah menempati
ruangan " + idRuangan + "!"
    output "Silakan cari ruangan lain untuk dokter " +
dokter.username + "."
    →

if assignDokterToRuangan(map, dokter.id, idRuangan) then
    output ""
    output "Dokter " + username + " berhasil diassign ke ruangan
" + idRuangan + "!"
else
    output ""

```

```
        output "Gagal mengassign dokter " + username + " ke ruangan "
+ idRuangan + "."
```

## 9. F18 - Exit

```
procedure keluar (input: char)

    repeat
        output("Apakah Anda sudah selesai menggunakan layanan ini?
(y/n) ")
        input(input)

        if input = 'y' or input = 'Y' then
            output("Terima kasih telah menggunakan sistem rumah sakit
Nimons. Sampai jumpa!")
            stop

        else if input = 'n' or input = 'N' then
            output("Terima kasih telah menggunakan sistem rumah sakit
Nimons. Sampai jumpa!")
            stop

        else
            output("Input tidak valid. Harap masukkan 'y' atau 'n'.")
            endif

    until FALSE
```

# HASIL PENGUJIAN PROGRAM

## 1. Inisialisasi Main Program

```
+-----+
|           SELAMAT DATANG           |
|           DI RUMAH SAKIT NIMONS     |
+-----+
              powered by Kelompok L
              K04 - STI

Masukkan jumlah ruangan: 5
Masukkan kapasitas ruangan: 3

Selamat datang! Silakan pilih opsi:
1. LOGIN
2. REGISTER PASIEN
4. LUPA PASSWORD
5. HELP
0. KELUAR
-----
```

Gambar 7.1 Inisialisasi Main Program

## 2. F01 - Login

#Kasus 1

Login sebagai Manager

Didefinisikan ketika inisialisasi list User langsung tercatat satu akun dengan role MANAGER username “admin” dan password “admin123”

```
-----
Pilihan Anda: 1
>>> LOGIN
Username : admin
Password : admin123
Login berhasil. Halo, admin!
Selamat pagi Manager admin!
```

Gambar 7.2 Login sebagai Manager

#### #Kasus 2

##### Login sebagai Dokter

Asumsi telah melakukan command TAMBAH DOKTER sebelum melakukan login

```
Pilihan Anda: 1
>>> LOGIN
Username : nico
Password : nico123
Login berhasil. Halo, nico!
Selamat pagi Dokter nico!

Halo, DOKTER nico!
-----
```

Gambar 7.2 Login sebagai Dokter

#### #Kasus 3

##### Login sebagai Pasien

Asumsi telah melakukan register

```
Pilihan Anda: 1
>>> LOGIN
Username : ami
Password : ami123
Login berhasil. Halo, ami!
Selamat pagi Pasien ami! Ada keluhan apa?

Halo, PASIEN ami!
-----
```

Gambar 7.3 Login sebagai Pasien

#### #Kasus 4

##### Tidak ada username terdaftar

```
Pilihan Anda: 1
>>> LOGIN
Username : ngarang
Password : ngarang123
Tidak ada Manager, Dokter, atau pun Pasien yang bernama ngarang!
```

Gambar 7.4 Tidak Ada Username Terdaftar

#Kasus 5

Kredensial salah

```
Pilihan Anda: 1
>>> LOGIN
Username : ami
Password : ami456
Login gagal. Username atau password salah.
```

Gambar 7.5 Kredensial Salah

### 3. F02 - Register

#Kasus 1

Register berhasil

```
>>> REGISTER
Username : ami
Password : ami123
Registrasi berhasil untuk ami.

Selamat datang! Silakan pilih opsi:
1. LOGIN
2. REGISTER PASIEN
4. LUPA PASSWORD
5. HELP
0. KELUAR
-----
Pilihan Anda: 1
>>> LOGIN
Username : ami
Password : ami123
Login berhasil. Halo, ami!
Selamat pagi Pasien ami! Ada keluhan apa?
```

Gambar 7.6 Register Berhasil

#Kasus 2

Register gagal

```
Pilihan Anda: 2
>>> REGISTER
Username : ami
Password : amilagi
Username ami sudah terdaftar.
```

Gambar 7.7 Register Gagal

#### 4. F03 - Logout

#Kasus 1

Logout berhasil

```
>>> LOGIN
Username : admin
Password : admin123
Login berhasil. Halo, admin!
Selamat pagi Manager admin!

Halo, MANAGER admin!
-----
Menu yang Tersedia:
1. LIHAT DENAH
2. LIHAT RUANGAN
5. HELP
10. TAMBAH DOKTER
11. ASSIGN DOKTER
9. LOGOUT
0. EXIT
-----
Pilihan Anda: 9
>>> LOGOUT
Logout berhasil. Sampai jumpa, admin!
```

Gambar 7.8 Logout

#### 5. F04 - Lupa Password

#Kasus 1

Ubah password berhasil

```
Testcase 1: Jeffreey
>>> LUPA_PASSWORD
Username: Jeffreey
Kode Unik: Je2fr2ey
Halo Dokter Jeffreey, silakan daftarkan ulang password anda!
Password Baru: JR1234
Password berhasil diubah!
>>> LOGIN
Username: Jeffreey
Password: JR1234
Selamat pagi Dokter Jeffreey!
```

Gambar 7.9 Password Berhasil Diubah

#Kasus 2  
User tidak terdaftar

```
Testcase 2: NONEXISTENT_USER
>>> LUPA_PASSWORD
Username: NONEXISTENT_USER
Username tidak terdaftar!
```

Gambar 7.10 User Tidak Terdaftar

#Kasus 3  
Kode unik tidak sesuai

```
Testcase 3: nimonsslatte
>>> LUPA_PASSWORD
Username: nimonsslatte
Kode Unik: nimonsslatte
Kode unik salah!
```

Gambar 7.11 Kode Unik Salah

## 6. F05 - Menu Help

#Kasus 1  
Help belum login

```
Pilihan Anda: 5
>>> HELP
===== HELP =====
Kamu belum login sebagai role apapun. Silahkan login terlebih dahulu.

1. LOGIN: Masuk ke dalam akun yang sudah terdaftar
2. REGISTER: Membuat akun baru

Footnote:
1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar
2. Jangan lupa untuk memasukkan input yang valid
```

Gambar 7.12 Help Belum Login



## #Kasus 2

### Help Manager

```
Pilihan Anda: 5
>>> HELP
===== HELP =====
Halo Manager admin. Kenapa kamu memanggil command HELP? Kan kamu manager, tapi yasudahla
h kamu pasti sedang kebingungan. Berikut adalah hal-hal yang dapat kamu lakukan:
1. LOGOUT: Keluar dari akun
2. TAMBAH_DOKTER: Menambahkan dokter baru
3. LIHAT_USER: Melihat seluruh user
4. EXIT: Keluar dari sistem

Footnote:
1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar
2. Jangan lupa untuk memasukkan input yang valid
```

Gambar 7.13 Help Manager

## #Kasus 3

### Help Dokter

```
Pilihan Anda: 5
>>> HELP
===== HELP =====
Halo Dokter nico. Kamu memanggil command HELP. Kamu pasti sedang kebingungan. Berikut ad
alah hal-hal yang dapat kamu lakukan:
1. LOGOUT: Keluar dari akun
2. DIAGNOSIS: Melakukan diagnosis pasien
3. NGOBATIN: Memberikan obat
4. EXIT: Keluar dari sistem

Footnote:
1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar
2. Jangan lupa untuk memasukkan input yang valid
```

Gambar 7.14 Help Dokter

## #Kasus 4

### Help Pasien

```
Pilihan Anda: 5
>>> HELP
===== HELP =====
Selamat datang, ami. Kamu memanggil command HELP. Kamu pasti sedang kebingungan. Berikut
adalah hal-hal yang dapat kamu lakukan:
1. LOGOUT: Keluar dari akun
2. DAFTAR_CHECKUP: Daftar untuk pemeriksaan
3. MINUM_OBAT: Minum obat
4. PULANGDOK: Meminta izin pulang
5. EXIT: Keluar dari sistem

Footnote:
1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar
2. Jangan lupa untuk memasukkan input yang valid
```

Gambar 7.15 Help Pasien

## 7. F06 - Lihat Denah

Inisialisasi jumlah ruangan dan kapasitas tiap ruangan

```
+-----+
|               |
|   SELAMAT DATANG   |
|   DI RUMAH SAKIT NIMONS   |
|               |
+-----+
```

powered by Kelompok L  
K04 - STI

Masukkan jumlah ruangan: 5  
Masukkan kapasitas ruangan: 3

Pilihan Anda: 1

>>> LIHAT\_DENAH

```
+---+---+---+---+---+
| 1 | 2 | 3 | 4 | 5 |
+---+---+---+---+---+
```

Gambar 7.16 Lihat Denah

```
-----
Pilihan Anda: 2
Masukkan ID Ruangan yang ingin dilihat : 1
>>> LIHAT_RUANGAN 1

--- Detail Ruangan 1 ---
Kapasitas : 3
Dokter    : Dokter gracie

Pasien di dalam ruangan :
    Tidak ada pasien di dalam ruangan saat ini.
-----
```

Gambar 7.17 Lihat Ruangan

## 8. F10 - Tambah Dokter

#Kasus 1

Berhasil tambah dokter oleh manager

```
Halo, MANAGER admin!  
-----  
Menu yang Tersedia:  
1. LIHAT DENAH  
2. LIHAT RUANGAN  
5. HELP  
10. TAMBAH DOKTER  
11. ASSIGN DOKTER  
9. LOGOUT  
0. EXIT  
-----  
Pilihan Anda: 10  
>>> TAMBAH_DOKTER  
Username: gracie  
Password: gracie123  
Registrasi berhasil untuk gracie.  
Username gracie sudah terdaftar.
```

Gambar 7.18 Berhasil Tambah Dokter

#Kasus 2

Tidak berhasil tambah dokter oleh manager

```
Pilihan Anda: 10  
>>> TAMBAH_DOKTER  
Username: gracie  
Username gracie sudah terdaftar sebagai dokter.
```

Gambar 7.19 Tidak Berhasil Tambah Dokter

#Kasus 1

Berhasil assign dokter ke ruangan

```
Pilihan Anda: 11  
>>> ASSIGN_DOKTER  
Username: gracie  
ID Ruangan : 1  
  
Dokter gracie berhasil diassign ke ruangan 1!
```

Gambar 7.18 Berhasil Assign Dokter ke Ruangan

#Kasus 2

Dokter sudah diassign ke ruangan lain sebelumnya

```
-----  
Pilihan Anda: 11  
>>> ASSIGN_DOKTER  
Username: gracie  
ID Ruangan : 2  
  
Dokter gracie sudah diassign ke ruangan 1!
```

Gambar 7.20 Dokter Sudah Diassign ke Ruangan Lain

#Kasus 3

Ruangan sudah ditempati

```
Pilihan Anda: 11  
>>> ASSIGN_DOKTER  
Username: dina  
ID Ruangan : 1  
  
Dokter gracie sudah menempati ruangan 1!  
Silakan cari ruangan lain untuk dokter dina.
```

Gambar 7.21 Ruangan Sudah Ditempati

#Kasus 4

Ruangan sudah ditempati dan dokter sudah diassign

```
Pilihan Anda: 11  
>>> ASSIGN_DOKTER  
Username: dina  
ID Ruangan : 1  
  
Dokter dina sudah ada di ruangan 2!  
Ruangan 1 juga sudah ditempati dokter gracie!
```

Gambar 7.22 Ruangan Sudah Ditempati dan Dokter Sudah Diassign

## 9. F18 - Exit

```
-----  
Pilihan Anda: 0  
Apakah Anda sudah selesai menggunakan layanan ini? (y/n) y  
Terima kasih telah menggunakan sistem rumah sakit Nimons. Sampai jumpa!
```

Gambar 7.23 Exit

## **LAMPIRAN**

[Form Asistensi](#)